Multi-Agent District Simulation

Alaa Bakhti
Thibaut Barroyer
Ilan Dubois
Xavier Yvonne

❖ Introduction

For our multi-agent system class project, we had to make the simulation of a neighborhood using the Repast library. We found this project interesting, because it allowed us to better understand what a multi-agent program can be, and it taught us to use Repast.

❖ How to use

The tests directory at the root of the project contains a file named 'big.map'. This file contains the map of the district to simulate. There are several elements in this map:

- H: House (capacity: 1 adult + 1 child)
- O: Office (capacity: 1000 persons)
- P: Park (capacity: 1 person / square)
- S: School (capacity: 1000 persons)
- B: Bus station

- ` `(Space): Nothing

List of Agents:

- Human
  - Adult
  - Child
- Bus
- Place

For our simulation, a day lasts 500 ticks.

All constants mentioned in this paragraph can be changed in the 'ContextCreator.java' file ; they are passed to other classes as arguments for constructors.

❖ Simulation logic

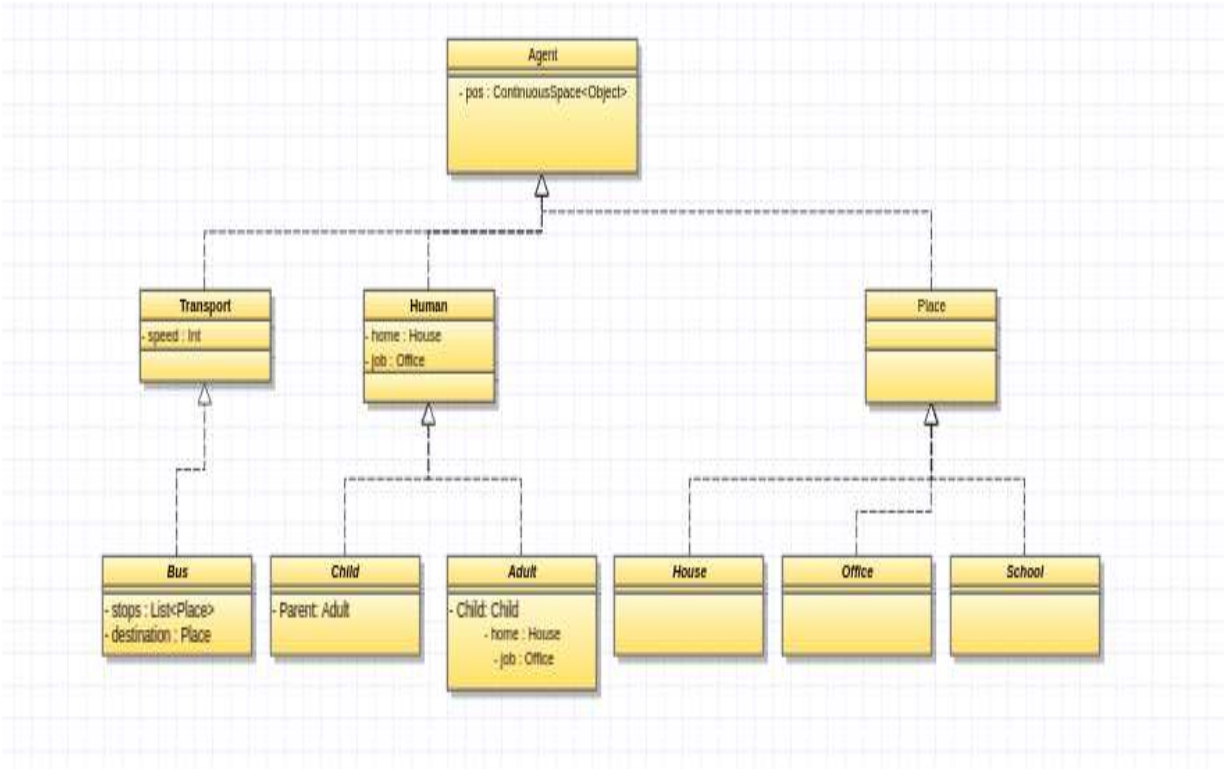
At initialization, every house is filled with a family composed of an adult and a child. To every adult is assigned a random office on the map. Every morning, adults bring their children to school, and then go to work. They work at the office, then go back to school to pick up their children, and finally they go back home. Two days a week, instead of going to office or school, they move to the park and have a random walk in it.

For every travel, every agent determines what is the fastest way to their destination. If they go to a station, they wait for the bus, and if walking to the next bus station reduces the duration of the journey, then they do it. This way, the agent makes some kind of Dijkstra's algorithm to know what is the best option to save time. This is comparable to the strategy real humans would use if they had a perfect knowledge of the transportation map and the time needed to wait at the bus station.

In this simulation, we can observe rush hours at the moment everyone goes to office or park. So we thought it was relevant that they all know transportation like office workers in a real district would. During weekends, activities are quite limited: only the park.

❖ Project architecture

Here is the UML diagram of our simulation:



As shown above, the simulation has 3 classes of agents. The first one is human that is composed with adults and children. Transportation is only composed by buses. If we want to add more transportation ways to the map, we can add more classes here. Last but not least, the Place class exposes all kinds of facilities represented on the map.

The ‘big.map’ file is read at initialization time line by line and counts all the needed information such as the number of houses, the size of the map, etc. After the instantiation of all agents, the file is read a second time to place everything at the right place on the map.

❖ Graphical contents

Here are the sprites used in the simulation:

-  : Adult
-  : Child
-  : House
-  : Office
-  : Park
-  : Bus

Those sprites, yet quite simple, allow a good visualization of the simulation.

❖ Planned upgrades

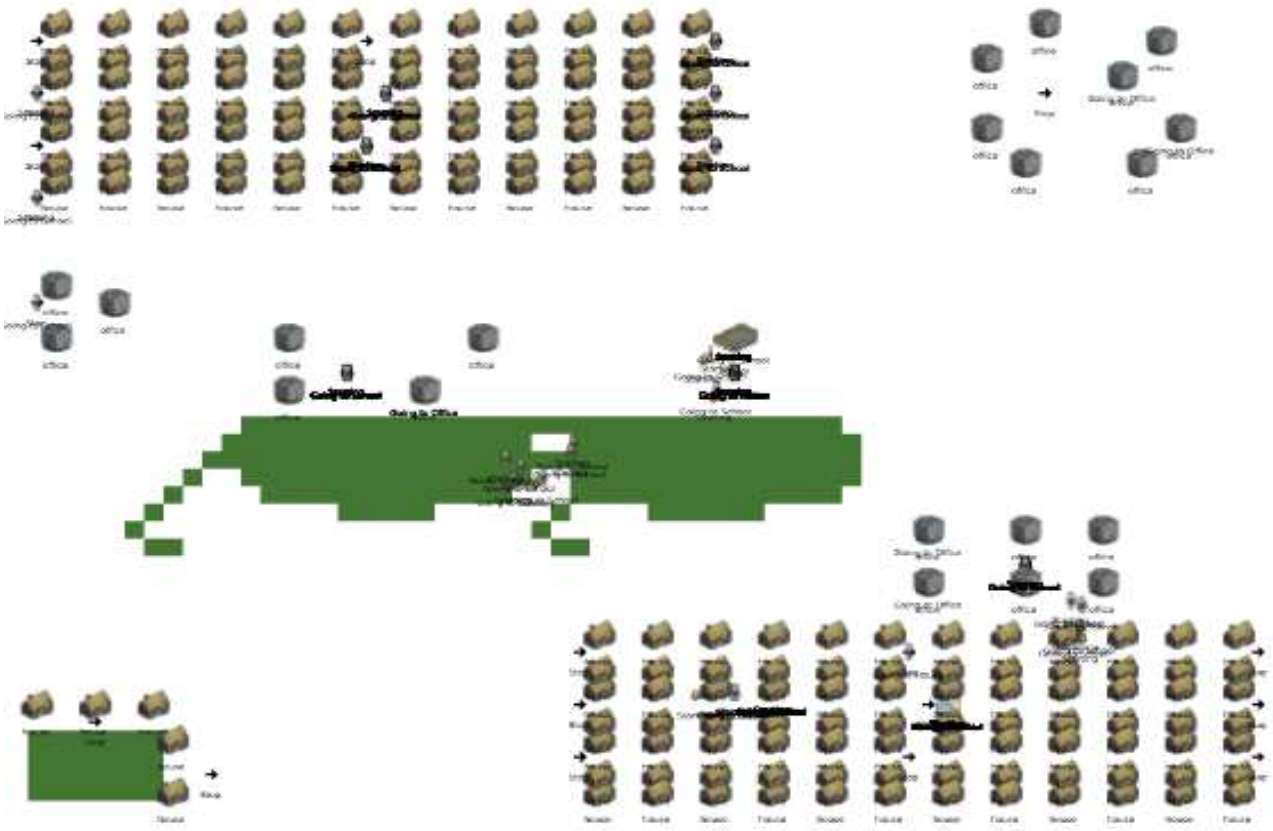
- Add a capacity for buses.
- Add new transportation ways: cars for adults and bikes for children.
- Children could grow and become adults. Adult could have children. Humans could possibly die.

- Humans could build new vehicles, houses and offices.
- Death could be natural (end of lifetime) or due to collisions between vehicles.
- Handle money: adults would earn money during work, give some to their children, and spend it at shops we could also add. There would also be a fare for the bus - broke people would have to walk.

❖ Difficulties met

- We had some conflicts between '.class' and 'Repast.settings' every time we wanted to add a class.
- We found the Repast interface difficult to install and inconvenient to use, even though it can produce fancy simulations.
- We wished the user could add their custom neighborhood map, but we didn't manage to implement this completely. This feature only works if the map is named 'big.map', is a valid file, and is put in the root directory like we did.
- We were lacking time to implement a complex simulation, due to the numerous projects we also had to deal with in the meantime.

❖ Simulation overview



❖ Conclusion

For this project, we have made a simple district multi-agent simulation. This simulation is interesting for observing the effects of rush hours and seeing how transportation is handled in real life.